Structured Operational Semantics for Graph Rewriting*

Andrei Dorman

Dip. di Filosofia, Università Roma Tre
LIPN – UMR 7030, Université Paris 13

`andrei.dorman@lipn.univ-paris13.fr`

Tobias Heindel

LIPN – UMR 7030, Université Paris 13

`tobias.heindel@lipn.univ-paris13.fr`

Process calculi and graph transformation systems provide models of reactive systems with labelled transition semantics. While the semantics for process calculi is compositional, this is not the case for graph transformation systems, in general. Hence, the goal of this article is to obtain a compositional semantics for graph transformation system in analogy to the structural operational semantics (SOS) for Milner's Calculus of Communicating Systems (CCS).

The paper introduces an SOS style axiomatization of the standard labelled transition semantics for graph transformation systems. The first result is its equivalence with the so-called Borrowed Context technique. Unfortunately, the axiomatization is not compositional in the expected manner as no rule captures "internal" communication of sub-systems. The main result states that such a rule is derivable if the given graph transformation system enjoys a certain property, which we call "complementarity of actions". Archetypal examples of such systems are interaction nets. We also discuss problems that arise if "complementarity of actions" is violated.

Key words: process calculi, graph transformation, structural operational semantics, compositional methods

1 Introduction

Process calculi remain one of the central tools for the description of interactive systems. The archetypal example of process calculi are Milner's π-calculus and the even more basic calculus of communication systems (CCS). The semantics of these calculi is given by labelled transition systems (LTS), which in fact can be given as a structural operational semantics (SOS). An advantage of SOS is their potential for combination with compositional methods for the verification of systems (see e.g. [17]).

Fruitful inspiration for the development of LTS semantics for other "non-standard" process calculi originates from the area of graph transformation where techniques for the derivation of LTS semantics from "reaction rules" have been developed [16, 7]. The strongest point of these techniques is the context independence of the resulting behavioral equivalences, which are in fact congruences. Moreover, these techniques have lead to original LTS-semantics for the ambient calculus [15, 3], which are also given as SOS systems. Already in the special case of ambients, the SOS-style presentation goes beyond the standard techniques of label derivation in [16, 7]. An open research challenge is the development of a general technique for the canonical derivation of SOS-style LTS-semantics. The problem is the "monolithic" character of the standard LTS for graph transformation systems.

In the present paper, we set out to develop a partial solution to the problem for what we shall call CCS-*like* graph transformation systems. The main idea is to develop an analogy to CCS where each action α has a co-action $\overline{\alpha}$ that can synchronize to obtain a silent transition; this is the so-called *communication rule*. In analogy, one can restrict attention to graph transformation systems with rules that allow to assign to each (hyper-)edge a unique *co-edge*. Natural examples of such systems are interaction nets as

---

introduced by Lafont [11, 1]. In fact, one of the motivations of the paper is to derive SOS semantics for interaction nets.

Structure and contents of the paper   We first introduce the very essentials of graph transformation and the so-called Borrowed Context (BC) technique [7] for the special case of (hyper-)graph transformation in Section 2. To make the analogy between CCS and BC as formal as possible, we introduce the system SOSBC in Section 3, which is meant to provide the uninitiated reader with a new perspective on the BC technique. Moreover, the system SOSBC emphasizes the "local" character of graph transformations as every transition can be decomposed into a "basic" action in some context. In particular, we do not have any counterpart to the communication rule of CCS, which shall be addressed in Section 4. We illustrate why it is not evident when and how two labeled transitions of two states that share their interface can be combined into a single synchronized action. However, we will be able to describe sufficient conditions on (hyper-)graph transformation systems that allow to derive the counterpart of the communication rule of CCS in the system SOSBC. Systems of this kind have a natural notion of "complementarity of actions" in the LTS.

2   Preliminaries

We first recall the standard definition of (hyper-)graphs and a formalism of transformation of hyper-graphs (following the double pushout approach). We also present the labelled transition semantics for hyper-graph transformation systems that has been proposed in [7]. In the present paper, the more general case of categories of graph-like structures is not of central importance. However, some of the proofs will use basic results of category theory.

Definition 2.1 (Hypergraphs and hypergraph morphisms)**.** Let Λ be a set of *labels* with associated *arity* function ar$\colon \Lambda \to \mathbb{N}$. A *($\Lambda$-labelled) hyper-graph* is a tuple $G = (E, V, \ell, \mathrm{cnct})$ where E is a set of *(hyper-)edges*, V is a set of *vertices* or *nodes*, $\ell \colon E \to \Lambda$ is the *labelling function*, and cnct is the *connection* function, which assigns to each edge $e \in E$ a string (e.g. a finite sequence) of *incident vertices* $\mathrm{cnct}(e) = v_1 \cdots v_n$ of length $\mathrm{ar}(\ell(e)) = n$ (where $\{v_1, \ldots, v_n\} \subseteq V$). Let $v \in V$ be a node; its *degree*, written $\deg(v)$ is the number of edges of which it is an incident node, i.e. $\deg(v) = |\{e \in E \mid v \text{ incident to } e\}|$ (where for any finite set M, the number of elements of M is $|M|$). We also write $v \in G$ and $e \in G$ if $v \in V$ and $e \in E$.

Let $G_i = (E_i, V_i, \ell_i, \mathrm{cnct}_i)$ $(i \in \{1,2\})$ be hyper-graphs; a *hyper-graph morphism* from G_1 to G_2, written $f \colon G_1 \to G_2$ is a pair of functions $f = (f_E \colon E_1 \to E_2, f_V \colon V_1 \to V_2)$ such that $\ell_2 \circ f_E = \ell_1$ and for each edge $e_1 \in E_1$ with attached nodes $\mathrm{cnct}(e) = v_1 \cdots v_n$ we have $\mathrm{cnct}_2(f_E(e)) = f_V(v_1) \cdots f_V(v_n)$. A hyper-graph morphism $f = (f_E, f_V) \colon G_1 \to G_2$ is *injective (bijective)* if both f_E and f_V are injective (bijective); it is an inclusion if both $f_E(e) = e$ and $f_V(v) = v$ hold for all $e \in E_1$ and $v \in V_1$. We write $G_1 \to G_2$ or $G_2 \leftarrow G_1$ if there is an inclusion from G_1 to G_2, in which case G_1 is a *sub-graph* of G_2.

To define double pushout graph transformation and the Borrowed Context technique [7], we will need the following constructions of hyper-graphs, which roughly amount to intersection and union of hyper-graphs.

Definition 2.2 (Pullbacks & pushouts of monos)**.** Let $G_i = (E_i, V_i, \ell_i, \mathrm{cnct}_i)$ $(i \in \{0,1,2,3\})$ be hyper-graphs and let $G_1 \to G_3 \leftarrow G_2$ be inclusions. The *intersection of G_1 and G_2* is the hyper-graph $G' = (E_1 \cap E_2, V_1 \cap V_2, \ell', \mathrm{cnct}')$ where $\ell'(e) = \ell_1(e)$ and $\mathrm{cnct}'(e) = \mathrm{cnct}_2(e)$ for all $e \in E_1 \cap E_2$. The *pullback of $G_1 \to G_3 \leftarrow G_2$* is the pair of inclusions $G_1 \leftarrow G' \to G_2$ and the resulting square is a pullback square (see Figure 1).

$$\begin{array}{ccc} G_3 \longleftarrow G_2 & \qquad & G_0 \longrightarrow G_2 \\ \uparrow \qquad \uparrow & & \downarrow \qquad \downarrow \\ G_1 \longleftarrow G' & & G_1 \longrightarrow G'' \end{array}$$
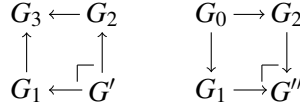
Figure 1: Pullback and pushout square

Let $G_1 \leftarrow G_0 \rightarrow G_2$ be inclusions; they are *non-overlapping* if both $E_1 \cap E_2 \subseteq E_0$ and $V_1 \cap V_2 \subseteq V_0$ hold. The *pushout of non-overlapping inclusions* $G_1 \leftarrow G_0 \rightarrow G_2$ is the pair of inclusions $G_1 \rightarrow G'' \leftarrow G_2$ where $G'' = (E_1 \cup E_2, V_1 \cup V_2, \ell'', \mathrm{cnct}'')$ is the hyper-graph that satisfies

$$\ell''(e) = \begin{cases} \ell_1(e) & \text{if } e \in E_1 \\ \ell_2(e) & \text{otherwise} \end{cases} \quad \text{and } \mathrm{cnct}''(e) = \begin{cases} \mathrm{cnct}_1(e) & \text{if } e \in E_1 \\ \mathrm{cnct}_2(e) & \text{otherwise} \end{cases}$$

for all $e \in E_1 \cup E_2$.

Finally, we are ready to introduce graph transformation systems and their labelled transition semantics.

Definition 2.3 (Rules and graph transformation systems). A *rule (scheme)* is a pair of non-overlapping inclusions of hyper-graphs $\rho = (L \leftarrow I \rightarrow R)$. Let A, B be hyper-graphs such that $A \leftarrow L$ and moreover $A \leftarrow I \rightarrow R$ is non-overlapping. Now, ρ *transforms A to B* if there exists a diagram as shown on the right such that the two squares are pushouts and there is an isomorphism $\iota \colon B' \rightarrow B$. A *graph transformation system* (GTS) is pair $\mathscr{S} = (\Lambda, \mathscr{R})$ where Λ is a set of labels and $\mathscr{R}$ is a set of rules.

$$\begin{array}{ccc} L \longleftarrow I \longrightarrow R \\ \downarrow \qquad \downarrow \qquad \downarrow \\ A \longleftarrow D \longrightarrow B' \end{array}$$

A graph transformation rule can be understood as follows. Whenever the left hand side L is (isomorphic to) a sub-graph of some graph A then this sub-graph can be "removed" from A, yielding the graph D. The vacant place in D is then "replaced" by the right hand side R of the rule. The middleman I is the memory of the connections L had with the rest of the graph in order for R to be attached in exactly the same place.

We now present an example that will be used throughout the paper to illustrate the main ideas.

Example 2.1. The system $\mathscr{S}_{ex} = (\Lambda, \mathscr{R})$ will be the following one in the sequel: $\Lambda = \{\alpha, \beta, \gamma, \dots\}$ such that $\mathrm{ar}(\alpha) = 2$, $\mathrm{ar}(\beta) = 3$ and $\mathrm{ar}(\gamma) = 1$; moreover $\mathscr{R}$ is the set of rules given in Figure 2 where the R_i represent different graphs (e.g. edges with labels R_i).

To keep the graphical representations clear, all inclusions in the running example are given implicitly by the spatial arrangement of nodes and edges.

(a) Rule "α/β"

(b) Rule "α/γ"

(c) Rule "β/γ"

(d) Rule "$\alpha/\beta/\gamma$"
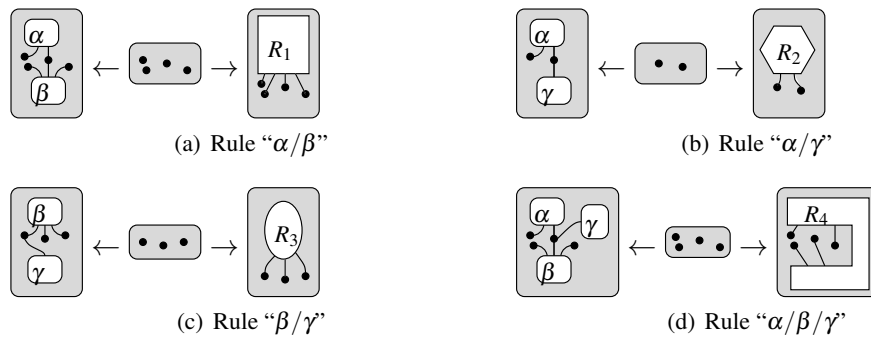
Figure 2: Reaction rules of $\mathscr{S}_{ex}$.

Remark 2.1 (Rule instances). Given a rule $L \leftarrow I \rightarrow R$ and a graph A such that $A \leftarrow L$, one can assume w.l.o.g. that $A \leftarrow I \rightarrow R$ is non-overlapping. The reason is that in each case, the rule $L \leftarrow I \rightarrow R$ could be replaced by an *isomorphic* "rule instance" $\rho' = L' \leftarrow I' \rightarrow R'$ (based on the standard notion of rule isomorphism).

In fact the result of each transformation step is unique (up to isomorphism). This is a consequence of the following fact.

Fact 2.4 (Pushout complements). *Let $G_2 \leftarrow G_1 \leftarrow G_0$ be a pair of hyper-graph inclusions where $G_i = (E_i, V_i, \ell_i, \mathrm{cnct}_i)$ ($i \in \{0, 1, 2\}$) such that for all $v \in V_1 \setminus V_0$ there does not exist any edge $e \in E_2 \setminus E_0$ such that v is incident to e. Then there exists a unique sub-graph $G_2 \leftarrow D$ such that (1) is a pushout square.*

$$
\begin{array}{ccc}
G_1 & \leftarrow & G_0 \\
\downarrow & & \downarrow \\
G_2 & \leftarrow & D
\end{array}
\quad (1)
$$

Definition 2.5 (Pushout Complement). Let $G_2 \leftarrow G_1 \leftarrow G_0$ be a pair of hyper-graph inclusions that satisfy the conditions of Fact 2.4; the unique completion $G_2 \leftarrow D \leftarrow G_0$ in (1) is the *pushout complement* of $G_2 \leftarrow G_1 \leftarrow G_0$.

Definition 2.6 (Labelled transition system). A *labelled transition system* (LTS) is a tuple $(S, \mathrm{Ł}, R)$ where S is a set of *states*, $\mathrm{Ł}$ is a set of *labels* and $R \subseteq S \times \mathrm{Ł} \times S$ is the *transition relation*. We write

$$
s \xrightarrow{\alpha} s'
$$

if $(s, \alpha, s') \in R$ and say that s can evolve to s' by performing α.

Definition 2.7 (DPOBC). Let $\mathscr{S} = (\Lambda, \mathscr{R})$ be a graph transformation system. Its LTS has all inclusions of hyper-graphs $J \rightarrow G$ as *states* where J is called the *interface*; the labels are all pairs of inclusions $J \rightarrow F \leftarrow K$, and a state $J \rightarrow G$ evolves to another one $K \rightarrow H$ if there is a diagram as shown on the right, which is called a DPOBC-*diagram* or just a BC-*diagram*. In this diagram, the graph D is called the *partial match of L*.

$$
\begin{array}{ccccccc}
D & \longrightarrow & L & \longleftarrow & I & \longrightarrow & R \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
G & \longrightarrow & G_c & \longleftarrow & C & \longrightarrow & H \\
\uparrow & & \uparrow & & \uparrow & & \nearrow \\
J & \longrightarrow & F & \longleftarrow & K
\end{array}
$$

For a technical justification of this definition, see [16], but let us give some intuitions on what this diagram expresses. States are inclusions, where the "larger" part models the whole "internal" state of the system while the "smaller" part, the interface, models the part that is directly accessible to the environment and allows for (non-trivial) interaction. As a particular simple example, one could have a Petri net where the set of places (with markings) is the complete state and some of the place are "open" to the environment such that interaction takes place by exchange of tokens.

The addition of agents/resources from the environment might result in "new" reactions, which have not been possible before. The idea of the LTS semantics for graph transformation is to consider (the addition of) "minimal" contexts that allow for "new" reactions as labels. The minimality requirement of an addition $J \rightarrow E$ or $J \rightarrow F$ is captured by the two leftmost squares in the BC diagram above: the addition $J \rightarrow F$ is "just enough" to complete part of the left hand side L of some rule. If the reaction actually takes place, which is captured by the other two squares in the upper row in the BC diagram, some agents might disappear / some resources might be used (depending on the preferred metaphor) and new ones might appear. Finally the pullback square in the BC diagram restricts the changes to obtain the new interface into the result state after reaction. As different rules might result in different deletion effects that are "visible" to the environment, the full label of each such "new" reaction is the "trigger" $J \rightarrow F$ together with the "observable" change $F \leftarrow K$ (with state $K \rightarrow H$ after interaction).

3   Three Layer SOS semantics

We start with a reformulation of the borrowed context technique that breaks the "monolithic" BC-step into axioms (that allow to derive the *basic actions*) and two rules that allow to perform these basic actions within suitable contexts. The axioms corresponds to the CCS-axioms that describe that the process $\alpha.P$ can perform the action α and then behaves as P, written $\alpha.P -\alpha\to P$ where α ranges over the actions $a, \bar{a}$, and τ. In the case of graphs, each rule $L \leftarrow I \rightarrow R$ gives rise to such a set of actions. More precisely, each subgraph D of L can be seen as an "action" with co-action $\widehat{D}^L \rightarrow L$ such that L is the union of D and $\widehat{D}^L$. For example, in the rule α/β, both edges α and β yield (complementary) basic actions.

Formally, in Table 1, we have the family of *Basic Action* axioms. It essentially represents all the possible uses of a transformation rule. In an (encoding of) CCS, the left hand side would be a pair of unary edges a and $\bar{a}$, which both disappear during reaction. Now, if only a is present "within" the system, it needs $\bar{a}$ to perform a reaction; thus, the part a of the left hand side induces the (inter-)action that consists in "borrowing" $\bar{a}$ and deleting both edges (and similarly for $\bar{a}$). In general, e.g. in the rule $\alpha/\beta/\gamma$ there might be more than two edges that are involved in a reaction and thus we have a whole family of actions. More precisely, each portion of a left hand side induces the action that consists in borrowing the missing part to perform the reaction (thus obtaining the coplete left hand side), followed by applying the changes that are described by the right part of the rule.

Next, we shall give counterparts for two CCS-rules that describe that an action can be performed in parallel to another process and under a restriction. More precisely, whenever we have the transition $P -\alpha\to P'$ and another process Q, then there is also a transition $P \parallel Q -\alpha\to P' \parallel Q$; similarly, we also have $(\nu b)P -\alpha\to (\nu b)P'$ whenever $\alpha \notin \{\bar{b}, b\}$. More abstractly, actions are preserved by certain contexts. The notion of context in the case of graph transformation, which will be the counterpart of process contexts such as $P \parallel [\cdot]$ and $(\nu b)[\cdot]$, is as follows.

Definition 3.1 (Context). A *context* is a pair of inclusions $C = J \rightarrow E \leftarrow J'$. Let $J \rightarrow G$ be a state (such that $E \leftarrow J \rightarrow G$ is non-overlapping); the *combination* of $J \rightarrow G$ with the context C, written $C[J \rightarrow G]$, is the inclusion of J' into the pushout of $E \leftarrow J \rightarrow G$ as illustrated in the following display.

$$\text{state: } \begin{matrix} G \\ \uparrow \\ J \end{matrix} \quad \text{context: } J \longrightarrow E \longleftarrow J' \quad \text{construction: } \begin{matrix} G \longrightarrow \overline{G} \\ \uparrow \quad \lrcorner \uparrow \quad \diagdown \\ J \longrightarrow E \longleftarrow J' \end{matrix} \quad \text{combination: } \begin{matrix} \overline{G} \\ \uparrow \\ J' \end{matrix}$$

The left inclusion of the context, i.e. $J \rightarrow E$, can also be seen as a state with the same interface. The pushout then gives the result of "gluing" E to the original G at the interface J; the second inclusion $J' \rightarrow E$ models a new interface, which possibly contains part of J and additional "new" entities in E.

With this general notion of context at hand, we shall next address the counterpart of name restriction, which we call *interface narrowing*, the second rule family in Table 1. In CCS, the restriction (νa) preserves only those actions that do not involve a. The counterpart of the context $(\nu a)[\cdot]$ is a context of the form $J \rightarrow J \leftarrow J'$. In certain cases, one can "narrow" a label while "maintaining" the "proper" action as made formal in the following definition.

Definition 3.2 (Narrowing). A *narrowing context* is a context of the form $C = J \rightarrow J \leftarrow J'$. Let $J \rightarrow F \leftarrow K$ be a label such that the pushout complement of $F \leftarrow J \leftarrow J'$ exists; then the *C-narrowing* of the label, written $C[J \rightarrow F \leftarrow K]$ is the lower row in the following display

$$C[J \rightarrow F \leftarrow K] := \begin{matrix} J \longrightarrow F \longleftarrow K \\ \uparrow \quad \lrcorner \uparrow \quad \ulcorner \uparrow \\ J' \longrightarrow F' \longleftarrow K' \end{matrix} \qquad \text{where } C = J \rightarrow J \leftarrow J'$$

where the left square is a pushout and the right one a pullback. Whenever we write $C[J \to F \leftarrow K]$, we assume that the relevant pushout complement exists.

If we think of the interface as the set of free names of a process, then restricting a name means removal from the interface. Thus, J' is the set of the remaining free names. If the pushout complement F' exists, it represents F with the restricted names erased. Finally, since a pullback here can be seen as an intersection, K' is K without the restricted names. So we finally obtain the "same" label where "irrelevant" names are not mentioned. It is of course not always possible to narrow the interface. For instance, one cannot restrict the names that are involved in labelled transitions of CCS-like process calculi. This impossibility is captured by the non-existence of the pushout complement.

With the notion of narrowing, we can finally define the interface narrowing rule in Table 1.

The final rule in Table 1 captures the counterpart of performing an action in parallel composition with another process P. In the case of graph transformation, this case is non-trivial since even the pure addition of context potentially interferes with the action of some state $J \to G$. For example, if an interaction involves the deletion of an (isolated) node, the addition of an edge to this node inhibits the reaction. However, for each transition there is a natural notion of non-inhibiting context; moreover, to stay close to the intuition that parallel composition with a process P only adds new resources and to avoid overlap with the narrowing rule, we restrict to monotone contexts.

Definition 3.3 (Compatible contexts). Let $C = J \to E \leftarrow \bar{J}$ be a context; it is *monotone* if $J \to \bar{J}$. Let $J \to F \leftarrow K$ be a label; now C is *non-inhibiting w.r.t.* $J \to F \leftarrow K$ if it is possible to construct the diagram (2) where both squares are pushouts. Finally, a context $J \to E \leftarrow \bar{J}$ is *compatible* with the label $J \to F \leftarrow K$ if it is non-inhibiting w.r.t. it and monotone.

$$\begin{array}{ccc} J & \longrightarrow F \longleftarrow & K \\ \downarrow & \downarrow & \downarrow \\ E & \longrightarrow E_1 \longleftarrow & E' \end{array} \quad (2)$$

In a label $J \to F \leftarrow K$, the left inclusion represents the addition of new entities that "trigger" a certain reaction. A compatible context is simply a context that is able to provide at least F, usually more than F, while not attaching new edges to nodes that disappear during reaction.

The last rule in the SOSBC-system of Table 1 is the embedding of a whole transition into a monotone context. To define this properly, we introduce a partial operation for the "combination" of co-spans (which happens to be a particular type of relative pushout of co-spans); this generalizes the narrowing construction.

Definition 3.4 (Cospan combination). Let $C = (J \to F \leftarrow K)$ and $\overline{C} = (J \to E \leftarrow \bar{J})$ be two cospans. They are *combinable* if there exists a diagram of the following form.

$$\begin{array}{ccc} J & \longrightarrow F \longleftarrow & K \\ \downarrow & \downarrow & \downarrow \\ E & \longrightarrow E_1 \longleftarrow & E' \\ \uparrow & \uparrow & \uparrow \\ \bar{J} & \longrightarrow \bar{F} \longleftarrow & \bar{K} =: \overline{C}[J \to F \leftarrow K] \end{array}$$

The label $\bar{J} \to \bar{F} \leftarrow \bar{K}$ is the *combination of C with $\overline{C}$*, and is denoted by $\overline{C}[J \to F \leftarrow K]$.

In fact, it is easy to show that compatible contexts are combinable with their label.

Lemma 3.5. *Given a reduction label $J \to F \leftarrow K$ and a compatible context $J \to E \leftarrow \bar{J}$ for it, we can split the diagram 2 in order to get*

$$
\begin{array}{ccc}
J \longrightarrow F \longleftarrow K & & J \longrightarrow F \longleftarrow K \\
\downarrow \quad \downarrow \quad \downarrow & & \downarrow \quad \downarrow \quad \downarrow \\
\bar{J} \longrightarrow \bar{F} \longleftarrow \bar{K} & \quad and \quad & E \longrightarrow E_1 \longleftarrow E' \cdot \\
\downarrow \quad \downarrow \quad \downarrow & & \uparrow \quad \uparrow \quad \uparrow \\
E \longrightarrow E_1 \longleftarrow E' & & \bar{J} \longrightarrow \bar{F} \longleftarrow \bar{K} = \bar{C}[J \rightarrow F \leftarrow K]
\end{array}
$$

With this lemma we can finally define the rule that corresponds to "parallel composition" of an action with another "process". Now the SOSBC-system does not only give an analogy to the standard SOS-semantics for CCS, we shall also see that the labels that are derived by the standard BC technique are exactly those labels that can be obtained from the basic actions by compatible contextualization and interface narrowing. In technical terms, the SOSBC-system of Table 1 is sound and complete.

- **Basic Actions**

$$
\frac{}{(D \rightarrow D) \xrightarrow{D \rightarrow L \leftarrow I} (I \rightarrow R)} \qquad \text{where} \quad \begin{array}{l} (L \leftarrow I \rightarrow R) \in \mathscr{S} \\ \text{and } D \rightarrow L \end{array}
$$

- **Interface Narrowing**

$$
\frac{(J \rightarrow G) \xrightarrow{J \rightarrow F \leftarrow K} (K \rightarrow H)}{(J' \rightarrow G) \xrightarrow{J' \rightarrow F' \leftarrow K'} (K' \rightarrow H)} \qquad \text{where} \quad \begin{array}{l} C = J \rightarrow J \leftarrow J' \\ \text{and } J' \rightarrow F' \leftarrow K' = C[J \rightarrow F \leftarrow K] \end{array}
$$

- **Compatible Contextualization**

$$
\frac{(J \rightarrow G) \xrightarrow{J \rightarrow F \leftarrow K} (K \rightarrow H)}{C[J \rightarrow G] \xrightarrow{C[J \rightarrow F \leftarrow K]} \bar{C}[K \rightarrow H]} \qquad \text{where} \quad \begin{array}{l} C = J \rightarrow E \leftarrow \bar{J} \text{ compatible with } J \rightarrow F \leftarrow K \\ \text{and } \bar{C} = (J \rightarrow F \leftarrow K)[C] \end{array}
$$

Table 1: Axioms and rules of the SOSBC-system.

Theorem 3.6 (Soundness and completeness). *Let $\mathscr{S}$ be a graph transformation system. Then there is a* BC-*transition*

$$
(J \rightarrow G) \xrightarrow{J \rightarrow F \leftarrow K} (K \rightarrow H)
$$

if and only if it is derivable in the SOSBC-*system.*

The main role of this theorem is not its technical "backbone", which is similar to many other theorems on the Borrowed Context technique. The main insight to be gained is the absence of any "real" communication between sub-systems; roughly, every reaction of a state can be "localized" and then derived from a basic action (followed by contextualization and narrowing). In particular, we do not have any counterpart to the communication-rule in CCS, which has complementary actions $P \xrightarrow{a} P'$ and $Q \xrightarrow{\bar{a}} Q'$ as premises and concludes the possibility of communication of the processes P and Q to perform the silent "internal" transition $P \parallel Q \xrightarrow{\tau} P' \parallel Q'$. The main goal is to provide an analysis of possible issues with a counterpart of this rule.

4   The composition rule for CCS-like systems

Process calculi, such as CCS and the π-calculus, have a so-called *communication rule* that allows to synchronize sub-processes to perform silent actions. The involved process terms have complementary actions that allow to interact by a "hand-shake". However, it is an open question how such a communication rule can be obtained for general graph transformations systems via the Borrowed Context technique. Roughly, the label of a transition does not contain information about which reaction rule was used to derive it; in fact, the same label might be derived using different rules. Intuitively, we do not know how to identify the two hands that have met to shake hands.

To elaborate on this using the metaphor of handshakes, assume that we have an agent that needs a hand to perform a handshake or to deliver an object. If we observe this agent reaching out for another hand, we cannot conclude from it which of the two possible actions will follow. In general, even after the action is performed, it still is not possible to know the decision of the agent – without extra information, which might however not be observable. However, with suitable assumptions about the "allowed actions", all necessary information might be available.

First, we recall from [2] that DPOBC-diagrams (as defined in Definition 2.7) can be composed under certain circumstances.

Fact 4.1. *Let*

$$(J \to G) \xrightarrow{J \to F \leftarrow K} (K \to H) \quad \text{and} \quad (J' \to G') \xrightarrow{J' \to F' \leftarrow K'} (K' \to H')$$

be two transitions obtained from two DPOBC-diagrams with the same rule $\rho = L \leftarrow I \to R$. Then, it is possible to build a DPOBC-diagram with the same rule for the composition of $J \to G$ and $J' \to G'$ along some common interface $J \leftarrow J_D^L \to J'$.

Take the following example as illustration of this fact.

Example 4.1 (Composition of transitions)**.** Let $J \to G$ be a state of $\mathscr{S}_{ex}$ that contains an edge α with its second connection in the interface as shown in Figure 3(a). Further, let $J' \to G'$ be a state that contains an edge β with its second connection in the interface as shown in Figure 3(b). Both graphs can trigger a reaction from rule $\alpha/\beta/\gamma$. Such a composition is shown in Figure 3(c).

Hence, we see that is in general possible to combine transitions to obtain new transitions. However, we emphasize at this point, that derivability of a counterpart of the communication rule of CCS is not the same question as the composition of pairs of transitions that come equipped with *complete* BC-diagrams. To clarify the problem, consider the following example where we cannot infer the used rule from the transition label.

Example 4.2. Let G be a graph composed of two edges α and β and consider a transition label where an edge γ is "added". Then it is justified by both rules α/γ and β/γ (see Figure 4).

We shall avoid this problem by restricting to suitable classes of graph transformation systems. Moreover, for simplicities sake, we shall focus on the derivation of "silent" transitions in the spirit of the communication rule of CCS.

Definition 4.2 (Silent label)**.** A label $J \to F \leftarrow K$ is *silent* or τ if $J = F = K$; a *silent transition* is a transition with a silent label.

Intuitively, a silent transition is one that does not induce any "material" change that is visible to an external observer that only has access to the interface of the states. Hence, in particular, a silent transition does not involve additions of the environment during the transition. Moreover, the interface remains
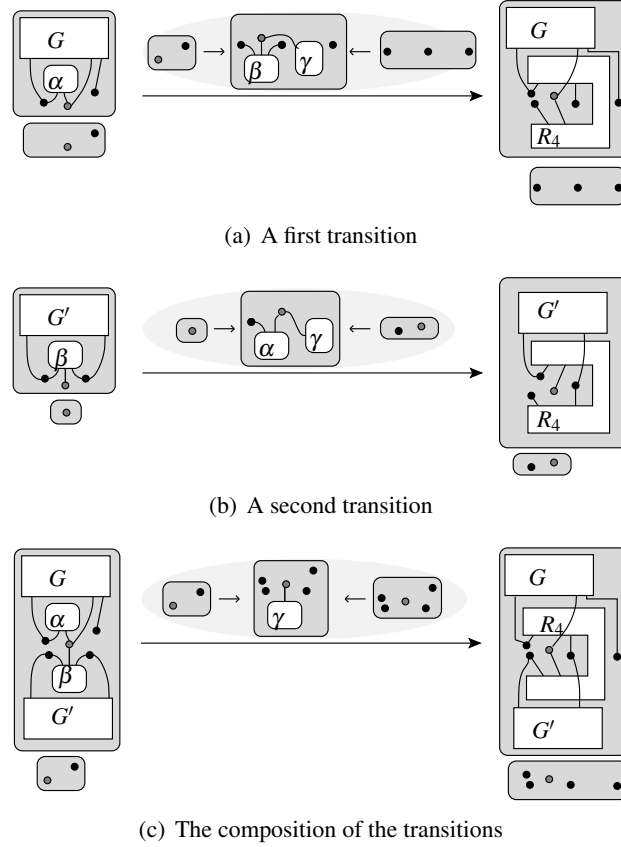
(a) A first transition

(b) A second transition

(c) The composition of the transitions

Figure 3: An example of composition.

unchanged. This latter requirement does not have any counterpart in process calculi, as the interface is given implicitly by the set of all free names. (In graphical encodings of process terms [3] it is possible to have free names in the interface even though there is no corresponding input or output prefix in the term.)

Now, with the focus on silent transitions, for a given rule $L \leftarrow I \rightarrow R$ we can illustrate the idea of complementary actions as follows. If a graph G contains a subgraph D of L and moreover a graph G' has the complementary subgraph of D in L in it, then G and G' can be combined to obtain a big graph $\overline{G}$ – the "parallel composition" of G and G' – that has the whole left hand side L as a subgraph and thus $\overline{G}$ can perform the reaction. A natural example for this are Lafont's interaction nets where the left hand side consist exactly of two hyper-edges, which in this case are called cells. The intuitive idea of complementary (basic) actions is captured by the notion of *active pairs*.

Definition 4.3 (Active pairs). For any inclusion $D \rightarrow L$, where $D \neq L$ and for all nodes v of D, $\deg(v) > 0$, let the following square be its initial pushout

$$\begin{array}{ccc} J_D^L & \longrightarrow & \widehat{D}^L \\ \downarrow & & \downarrow \\ D & \longrightarrow & L \end{array},$$

i.e. $\widehat{D}^L$ is the smallest subgraph of L that allows for completion to a pushout. We call $\widehat{D}^L$ the *complement of D in L* and J_D^L the *minimal interface of D in L* and we write $\{D, D'\} \equiv L$ if $D' = \widehat{D}^L$. The set of *active*

(a) A transition from rule α/γ

(b) A transition from rule β/γ
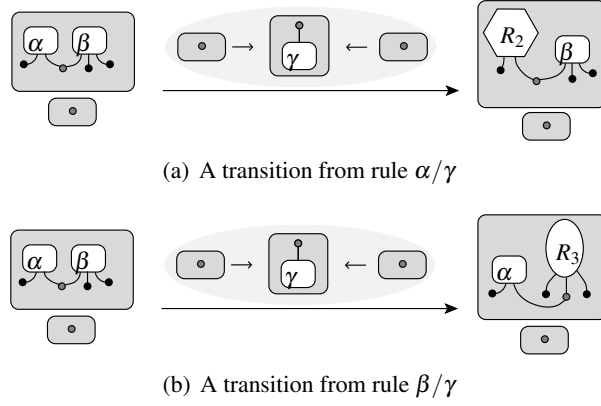
Figure 4: Same transition label for different rules.

pairs is

$$\mathbb{D} = \big\{\ \{D, \widehat{D}^L\}\ \mid L \leftarrow I \rightarrow R \in \mathcal{R}, D \rightarrow L,\ D \neq L,\ \forall v \in D.\deg(v) > 0\big\}.$$

Abusing notation, we also denote by $\mathbb{D}$ the union of $\mathbb{D}$.

It is easy to verify that the complement of $\widehat{D}^L$ in L is D itself and that its minimal interface is also J_D^L. It is the set of "acceptable" partial matches in the sense that they do not yield a τ-reaction on their own. Indeed, if D is equal to L, then the resulting transition of this partial match is a τ-transition. And if it is just composed of vertices, its complement is L and thus not acceptable.

Example 4.3 (Active pairs). In our running example, the set $\mathbb{D}$ of our example is in obvious bijection to

$$\big\{\{\alpha, \beta\}, \{\alpha, \gamma\}, \{\beta, \gamma\}, \{\alpha, \beta + \gamma\}, \{\alpha + \beta, \gamma\}, \{\alpha + \gamma, \beta\}\big\}.$$

The minimal interface of any pair is a single vertex.

This completes the introduction of preliminary concepts to tackle the issues that have to be resolved to obtain "proper" compositionality of transitions.

4.1 Towards a partial solution

Let us address the problem of identifying the rule that is "responsible" for a given interaction. We start by considering the left inclusions of labels, which intuitively describe possible borrowing actions from the environment. Relative to this, we define the *admissible rules* as those rules that can be used to let states evolve while borrowing the specified "extra material" from the environment.

Definition 4.4 (Admissible rule). Let $J \rightarrow G$ be a state and let $J \rightarrow F$ be an inclusion (which represents a possible contribution of the context). A rule ρ is *admissible (for $J \rightarrow F$)* if $L \not\rightarrow G$ and it is possible to find $D \in \mathbb{D}$ and L the left-hand side of ρ, such that the following diagram commutes

$$
\begin{array}{ccc}
 & & L \leftarrow \\
 & \diagup\!\!\!\!\!\times & \downarrow \\
G & \longrightarrow & G^c \\
\uparrow & & \uparrow \\
J & \longrightarrow & F \\
\uparrow & & \uparrow \\
J_D^L & \longrightarrow & D
\end{array}
$$

where $J_D^L \to D$ is the minimal interface of D in L. We call D the *rule addition*.

This just means that G *can* evolve using the rule ρ if D is added at the proper location.

Proposition 4.5 (Precompositionality). *Let $J \to G \xrightarrow{J \to F \leftarrow K} K \to H$ and $J' \to G' \xrightarrow{J' \to F' \leftarrow K'} K' \to H'$ be two transitions such that a single rule ρ is admissible for both, and let D and D' be their respective rule additions. If $\{D, D'\} \in \mathbb{D}$, it is possible to compose G and G' into a graph $\overline{G}$ in a way to be able to derive a τ-transition using rule ρ.*

Proof. We first show that in such a case, $D' \to G$ and the pushout of $G \leftarrow D' \to L$ is exactly G^c. Similarly, $D \to G'$ and the pushout of $G' \leftarrow D \to L$ is exactly G'^c. Then, it is easy to see that it is possible to build the DPOBC-diagram D_1 using rule ρ on G (respectively G') yelding the transition $(J \to G) \xrightarrow{J \to F \leftarrow K_1} (K_1 \to H_1)$ for some K_1, H_1 (respectively the DPOBC-diagram D_2 yelding the transition $(J \to G) \xrightarrow{J \to F \leftarrow K_2} (K_2 \to H_2)$ for some K_2, H_2), and then compose D_1 and D_2.

This follows from $\{D, D'\} \in \mathbb{D}$ and $\overline{G} \equiv G^c$. Indeed, $\overline{E} = L$ so the top left morphism of the composed DPOBC-diagram is an isomorphism and so are the ones under it, using basic pushout properties. $\square$

This first result motivates the following definition.

Definition 4.6 (τ-compatible). In the situation of Proposition 4.5, we say the two transitions are τ-*compatible*.

Remark 4.1. In general, in Proposition 4.5, the result of the τ-transition cannot be constructed from H and H'; thus we do not yet speak of compositionality.

Example 4.4. Let G be a graph composed of two edges α and γ and G' of two edges β and γ (see Figure 5). Then the rule α/β is admissible for both transitions and moreover they are τ-compatible. The rule α/β yields the respective rule additions. "Glueing" G and G' by their interface results in a graph with edges α, β and two γs; the latter graph can perform a τ-reaction from rule α/β, which however does not give the desired result since the target state is not the "expected composition" of H and H'. In other words, although we have been able to construct a τ-transition, it is not the composition of the original transitions.

(a) A transition from rule β/γ

(b) A transition from rule α/γ
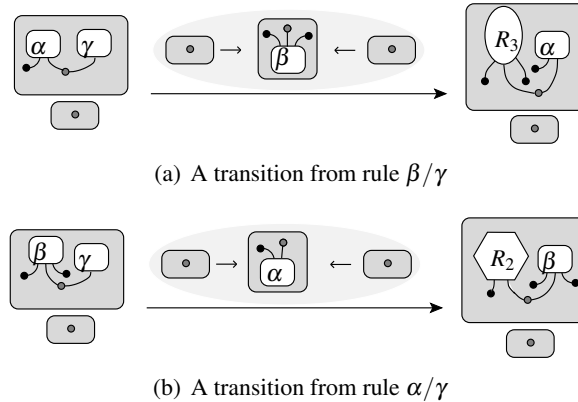
Figure 5: τ-compatible, but not composable: different rules.

We can see from the examples here that the difficulty of defining a composition of transitions comes mainly from three facts. The first is that a partial match can have several subgraphs triggering a reaction. This is delt with by the construction of the set of active pairs. The second one is the possibility to connect multiple edges together, not knowing which one exactly is consumed in the reaction. Finally, a given edge can have multiples ways of triggering a reaction.

4.2   Sufficient conditions

We now give two frameworks in which neither of the two last problems do occur. Avoiding each of them separately is enough to define compositionality properly. Both cases are inspired by the study of interaction net systems [12, 6, 14], which can be represented in the obvious manner as graph transformation systems. In these systems, the DPOBC-diagram built from an admissible rule of a transition is necessarily the one that has to be used to derive the transition. In one case, it works for essentially the same reasons as in CCS: every active element can only interact with a unique other element, such as a vs. $\bar{a}$, b vs. $\bar{b}$. In the other one, the label itself is not enough, but since we also know where it "connects" to the graph, it is possible to "find" the partner that was involved in the transition.

   We introduce interaction graph systems, which are caracterized among other rewriting systems by the form of the left-hand sides of the reaction rules, composed of exactly two hyperedges connected by a single node. We fix a labeling alphabet Λ.

Definition 4.7. An *activated pair* is a hypergraph L on Λ composed of two hyperedges e and f and a node v such that v appears exactly once in $\mathrm{cnct}(e)$ and once in $\mathrm{cnct}(f)$. If v is the i-th incident vertex of e labelled α and the j-th incident vertex of f labelled β, we denote the activated pair by $e_i \bowtie f_j$ and label it by $\alpha_i \bowtie \beta_j$.

   An *interaction graph system* $(\Lambda, \mathscr{R})$ is given by a set of reaction rules $\mathscr{R}$ over hypergraphs on Λ where all left-hand side of rules are activated pairs, and nodes are never deleted, i.e. for any rule $\rho = L \leftarrow I \rightarrow R$,

- L is an activated pair;

- for any node v, $v \in L \Rightarrow v \in I$.

   Note that for any interaction graph system, the set $\mathbb{D}$ is composed of pairs $\{D, D'\}$ where each of them is composed of an edge and its connected vertices. Also the minimal interface of any active pair $\{D, D'\}$ is a single node. It is also the case that it is enough for interfaces to be composed of vertices only.

Example 4.5. SIMPLY WIRED HYPERGRAPHS Lafont interaction nets are historically the first interaction nets. They appear as an abstraction of linear logic proof-nets [12]. Originally, Lafont nets have several particular features, but the one we are interested in is the condition on connectivity.

Definition 4.8. Let $N = (E, V, \ell, \mathrm{cnct})$ be a hypergraph on Λ.
   The graph N is *simply wired* if $\forall v \in V, \deg(v) \leq 2$. When $\deg(v) = 1$, we say that v is *free*.

   In other words, vertices are only incident to at most two edges of a graph. Note that in this special case no issues arise if we restrict to the sub-category of simply wired hypergraphs. For this, we argue that the purpose of the interface is the possible addition of extra context; thus, in simply wired hypergraphs, it is meaningless for a vertex that is already connected to two edges to be in the interface.

Definition 4.9 (Lafont interaction graph system)**.** A *Lafont interaction graph* is a simply connected graph such that its interface consists of free vertices only. A *Lafont system* $\mathbb{L} = (\Lambda, \mathscr{R})$ is given by reaction rules over Lafont interaction graphs; it is *partitioned* if two left-hand sides only overlap trivially, i.e. for two rules $\rho_j = L_j \leftarrow I_j \rightarrow R_j \in \mathscr{R}$ ($j = 1, 2$), either $L_1 = L_2$ or $L_1 \cap L_2$ is the empty graph (without any nodes and any hyperedges).

Lemma 4.10. *Let $\mathbb{L}$ be a partitioned Lafont system, let $J \rightarrow G$ be a state, let $(J \rightarrow G) \xrightarrow{J \rightarrow F \leftarrow K} (K \rightarrow H)$ be a non-τ transition. Then there is exactly one admissible rule for this transition.*

Example 4.6. HYPERGRAPHS WITH UNIQUE PARTNERS By generalizing Lafont interaction nets, we obtain so called *multiwired* interaction nets. But then we lose the unicity of the rule for a given transition label. It can be recovered by another condition.

Definition 4.11 (Unique partners). Let $\mathbb{I} = (\Lambda, \mathscr{R})$ be an interaction graph system. We say it is *with unique partners* if for any $\alpha \in \Lambda$ and for all $i \le \mathtt{ar}(\alpha)$, there exists a unique $\beta \in \Lambda$ and a unique $j \le \mathtt{ar}(\beta)$ such that $\alpha_i \bowtie \beta_j$ is the label of a left-hand side of a rule in $\mathscr{R}$.

Lemma 4.12. *Let $J \to G$ a state of $\mathbb{I}$ and $(J \to G) \xrightarrow{J \to F \leftarrow K} (K \to H)$ a non-τ reaction label. Then there is exactly one admissible rule ρ for this transition.*

Finally, we conclude our investigation with the following positive result.

Theorem 4.13 (Compositionality). *Let $(\Lambda, \mathscr{R})$ be a Lafont interaction graph system, or an interaction graph system with unique partners. Let $\mathbb{D}$ be its set of active pairs.*

Let $t_1 = (J \to G) \xrightarrow{J \to F \leftarrow K} (K \to H)$ and $t_2 = (J' \to G') \xrightarrow{J' \to F' \leftarrow K'} (K' \to H')$ be two non-τ transitions and D and D' their respective rule additions.

If $\{D, D'\} \equiv L \in \mathbb{D}$, let $\overline{G}$ and $\overline{H}$ are described by the following diagrams
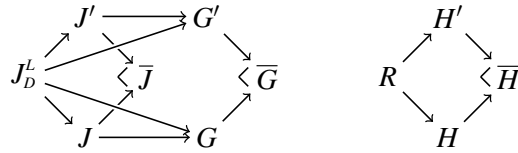
where $J_D^L \to J$ and $J_D^L \to J'$ are the inclusions from the admissibility of ρ for states $J \to G$ and $J' \to G'$ (Definition 4.4).

Then

$$(\overline{J} \to \overline{G}) \xrightarrow{\overline{J} \to \overline{J} \leftarrow \overline{J}} (\overline{J} \to \overline{H}).$$

Sketch of proof. By Lemma 4.10 or 4.12, there exists exactly one rule $\rho \in \mathscr{R}$ with L as a left-hand side that allows to derive transitions t_1 and t_2 – it is indeed the same rule for both. Let $\mathtt{D}$ be the composition diagram of the DPOBC-diagrams justifying the transitions.

It is first shown that $\overline{G} \equiv \overline{G}_c$. Since the upper and lower left squares of $\mathtt{D}$ are pushouts we can infer that $\overline{D} \equiv L$ and $\overline{J} \equiv \overline{F}$. Finally, since no vertex is deleted (see Definition 4.7), we have $\overline{J} \to \overline{C}$ and thus $\overline{K} \equiv \overline{J}$.

So $\mathtt{D}$ is a BC-diagram of a τ-reaction from $\overline{J} \to \overline{G}$ to $\overline{J} \to \overline{H}$.

$\square$

In fact, the main property that we have used is the following.

Definition 4.14 (Complementarity of Actions). A graph transformation systems satisfies *Complementarity of Actions* if for each transition $(J \to G) \xrightarrow{J \to F \leftarrow K} (K \to H)$ there is a unique rule $L \leftarrow I \to R$ such that there exists a DPOBC-diagram as shown to the right.
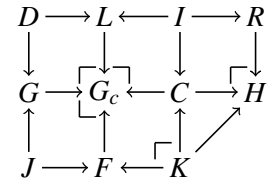
In this situation, we can effectively determine if two transitions are τ-compatible. Thus we can derive a counterpart of the communication rule of CCS. Hence, if a graph transformation systems satisfies Complementarity of Actions then a rule of the following form is derivable in SOSBC.

$$\frac{t = (\overline{J} \to G) \xrightarrow{\overline{J} \to \overline{F} \leftarrow \overline{K}} (\overline{K} \to H) \qquad t' = (\overline{J} \to G') \xrightarrow{\overline{J} \to \overline{F}' \leftarrow \overline{K}'} (\overline{K}' \to H')}{(\overline{J} \to \overline{G}) \xrightarrow{\overline{J} \to \overline{J} \leftarrow \overline{J}} (\overline{J} \to \overline{H})} \quad t \text{ and } t' \text{ } \tau\text{-compatible}$$

In other words, in a graph transformation system with Complementarity of Actions we can apply the results of [2] to obtain a counterpart to the communication rule.

5 Related and Future work

On a very general level, the present work is meant to strengthen the conceptual similarity of graph transformation systems and process calculi; thus it is part of a high-level research program that has been the theme of a Dagstuhl Seminar in 2005 [9]. In this wide field, structural operational semantics is occasionally considered as an instance of the tile model (see [8] for an overview). With this interpretation, SOS has served as motivation for work on operational semantics of graph transformation systems (e.g. [5]).

A new perspective on operational semantics, namely the "automatic" generation of labeled transition semantics from reaction rules, has been provided by the seminal work of Leifer and Milner [13] and its successors [16, 7]; as an example application, we want to mention the "canonical" operational semantics for the ambient calculus [15]. The main point of the latter work is the focus on the "properly" *inductive* definition of structural operational semantics. To the best of our knowledge, there is no recent work on the operational semantics of graph transformation systems that provides a general method for the inductive definition of *operational semantics*. This is not to be confused with the inductive definition of graphical encodings of process calculi on (global) states.

With this narrower perspective on techniques for the "automatic" generation of LTSs, we want to mention that some ideas of our three layer semantics in Section 3 can already be found in [3], where all rules of the definition of the labelled transition semantics have at most one premise. This is in contrast to the work of [15] where the labelled transition semantics is derived from two smaller subsystems: the process view and the context view; the subsystems are combined to obtain the operational semantics. The latter work is term based and it manipulates complete subterms of processes using the lambda calculus in the meta-language. We conjecture that the use of this abstraction mechanism is due to the term structure of processes.

Concerning future work, the first extension of the theory concerns more general (hierarchical) graph-like structures as captured by adhesive categories [10] and their generalizations (e.g. [4]). Moreover, as an orthogonal development, we plan to consider the case of more general rules that are allowed to have an arbitrary (graph) morphism on the right hand side; moreover, also states are arbitrary morphisms. The general rule format is important to model substitution in name passing calculi while arbitrary graph morphisms as states yield more natural representations of (multi-wire) interaction nets. The main challenge is the quest for more general sufficient conditions that allow for non-trivial compositions of labelled transitions, which can be seen as a general counterpart of the CCS communication rule.

6 Conclusion

We have reformulated the BC technique as the SOSBC-system in Table 1 to make a general analogy to the SOS-rules for CCS. There is no need for a counterpart of the communication rule. We conjecture that this is due to the "flat" structure of graphs as opposed to the tree structure of CCS-terms.

The main contribution concerns questions about the derivability of a counterpart of the communication rule. First, we give an example, which illustrates that the derivability of such a rule is non-trivial; however, it is derivable if the relevant graph transformation system satisfies *Complementarity of Actions*. We have given two classes of examples that satisfy this requirement, namely hyper-graphs with unique partners and simply wired hyper-graphs. This is a first step towards a "properly" inductive definition of structural operational semantics for graph transformation systems.

with drafts and ideas about a more general research program on compositionality in graph transformation. We are also grateful for the constructive criticism and the helpful comments of the anonymous referees.

References

[1] V. Alexiev (1999): *Non-deterministic interaction nets*. Ph.D. thesis, University of Alberta, Edmonton, Alta., Canada.

[2] P. Baldan, H. Ehrig & B. König (2006): *Composition and Decomposition of DPO Transformations with Borrowed Context*. In: *Proc. of ICGT '06 (International Conference on Graph Transformation)*, Springer, pp. 153–167, doi:10.1007/11841883_12. LNCS 4178.

[3] F. Bonchi, F. Gadducci & G. V. Monreale (2009): *Labelled transitions for mobile ambients (as synthesized via a graphical encoding)*. Electronic Notes in Theoretical Computer Science 242(1), pp. 73–98, doi:10.1016/j.entcs.2009.06.014.

[4] B. Braatz, H. Ehrig, G. Karsten & U. Golas (2010): *Finitary M-adhesive categories*. In: *Graph Transformations: 5th International Conference, ICGT 2010, Twente, the Netherlands, September 27–October 2, 2010, Proceedings*, Springer-Verlag, pp. 234–249, doi:10.1007/978-3-642-15928-2_16.

[5] Andrea Corradini, Reiko Heckel & Ugo Montanari (2000): *Graphical Operational Semantics*. In: *ICALP Satellite Workshops*, pp. 411–418.

[6] T. Ehrhard & L. Regnier (2006): *Differential interaction nets*. Theoretical Computer Science 364(2), pp. 166–195, doi:10.1016/j.tcs.2006.08.003.

[7] H. Ehrig & B. König (2006): *Deriving Bisimulation Congruences in the DPO Approach to Graph Rewriting with Borrowed Contexts*. Mathematical Structures in Computer Science 16(6), pp. 1133–1163, doi:10.1017/S096012950600569X.

[8] F. Gadducci & U. Montanari (2000): *The tile model*. In Gordon D. Plotkin, Colin Stirling & Mads Tofte, editors: *Proof, Language, and Interaction*, The MIT Press, pp. 133–166.

[9] B. König, U. Montanari & P. Gardner, editors (2005): *04241 Abstracts Collection*. Dagstuhl Seminar Proceedings 04241, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany. Available at `http://drops.dagstuhl.de/opus/volltexte/2005/27`.

[10] S. Lack & P. Sobociński (2005): *Adhesive and quasiadhesive categories*. RAIRO - Theoretical Informatics and Applications 39(2), pp. 522–546, doi:10.1051/ita:2005028.

[11] Y. Lafont (1990): *Interaction nets*. In: *Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '90, ACM, New York, NY, USA, pp. 95–108, doi:10.1145/96709.96718.

[12] Y. Lafont (1995): *From proof-nets to interaction nets*. In: *Proceedings of the workshop on Advances in linear logic*, Cambridge University Press, New York, NY, USA, pp. 225–247, doi:10.1017/CBO9780511629150.012.

[13] J. J. Leifer & R. Milner (2000): *Deriving Bisimulation Congruences for Reactive Systems*. In Catuscia Palamidessi, editor: *CONCUR*, Lecture Notes in Computer Science 1877, Springer, pp. 243–258, doi:10.1007/3-540-44618-4_19.

[14] D. Mazza (2006): *Interaction Nets: Semantics and Concurrent Extensions*. Ph.D. thesis, Université de la Méditerranée & Roma Tre.

[15] J. Rathke & P. Sobociński (2010): *Deriving structural labelled transitions for mobile ambients*. Information and Computation 208, pp. 1221–1242, doi:10.1016/j.ic.2010.06.001.

[16] V. Sassone & P. Sobociński (2003): *Deriving Bisimulation Congruences Using 2-categories*. Nordic Journal of Computing 10(2), pp. 163–183.

[17] A. Simpson (2004): *Sequent calculi for process verification: Hennessy-Milner logic for an arbitrary GSOS*. Journal of Logic and Algebraic Programming 60–61, pp. 287–322, doi:10.1016/j.jlap.2004.03.004.